Augmented molecular graph kernel QSARs

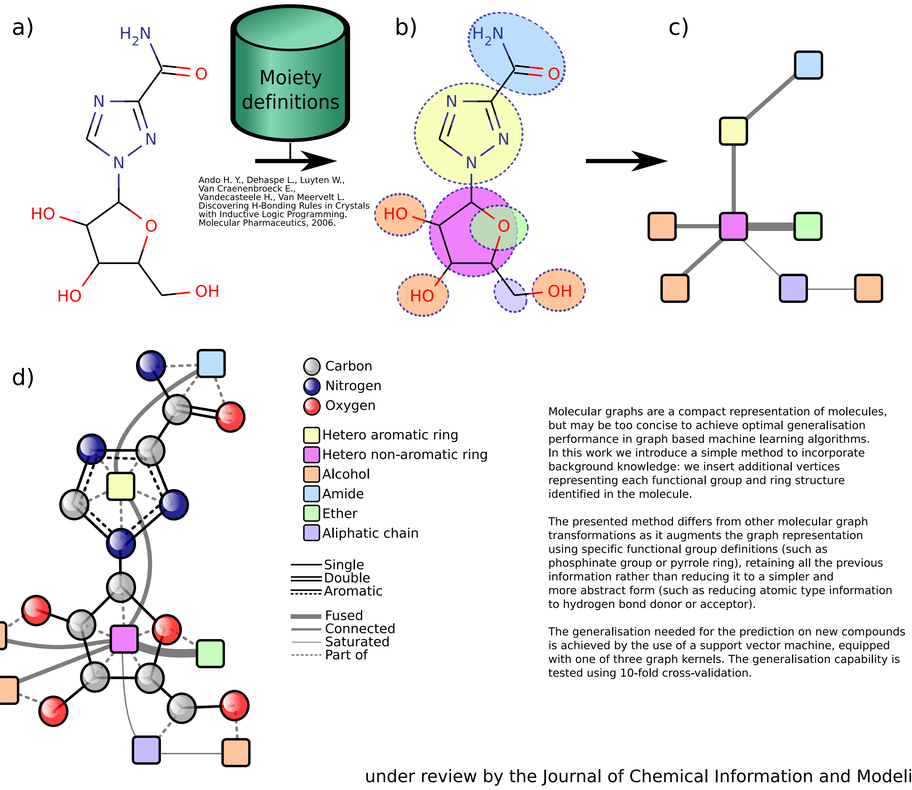
KATHOLIEKE UNIVERSITEIT LEUVEN

DTAI MACHINE LEARNING

Kurt De Grave
kurt.degrave@cs.kuleuven.be

Fabrizio Costa
fabrizio.costa@cs.kuleuven.be

Molecular graph augmentation



a) structure of ribavirin (example input)
b) identification of functional groups and rings
c) the moieties are encoded as extra nodes, which are added to the original structure to obtain
d) the augmented graph.

Molecular graphs are a compact representation of molecules, but may be too concise to achieve optimal generalisation performance in graph based machine learning algorithms. In this work we introduce a simple method to incorporate background knowledge: we insert additional vertices representing each functional group and ring structure identified in the molecule.

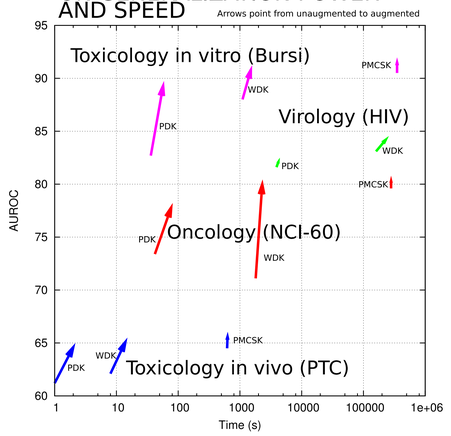
The presented method differs from other molecular graph transformations as it augments the graph representation using specific functional group definitions (such as phosphinate group or pyrrole ring), retaining all the previous information rather than reducing it to a simpler and more abstract form (such as reducing atomic type information to hydrogen bond donor or acceptor).

The generalisation needed for the prediction on new compounds is achieved by the use of a support vector machine, equipped with one of three graph kernels. The generalisation capability is tested using 10-fold cross-validation.

Procedure

- ► Add a vertex for each identified moiety (77 types)
- ► Connect all constituent atom vertices to the moiety vertex (part-of link)
- ► Connect moiety vertices if the moieties share one or more vertices (fused link) or if there is a bond between them (connected link)
- ► If a moiety is connected to an aliphatic chain, the link is labeled respectively as saturated or unsaturated

EFFECT OF AUGMENTATION ON GENERALIZATION POWER AND SPEED



Existing graph kernels

PAIRWISE DISTANCE KERNEL (PDK)

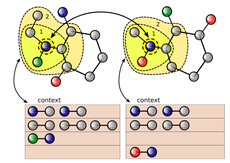
Borgwardt K. M. and Kriegel H.-P.
Shortest-Path Kernels on Graphs. In Proc. ICDM, pp.74-81, 2005.

The idea is to compute the similarity between two graphs by comparing all the respective pairs of vertices annotated with their topological distances. This is achieved by 1) first calculating the shortest path distance between all pairs of vertices using Floyd-Warshall's algorithm, and 2) then computing an all-pairs-shortest-paths kernel on edge walks of length 1 on an appropriately modified graph.

WEIGHTED DECOMPOSITION KERNEL (WDK)

Menchetti S., Costa F., and Frasconi P.
Weighted decomposition kernels. In Proc. ICML, pp. 585-592, 2005.

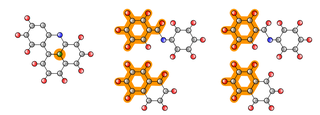
In the WDK, the neighborhood of a given radius is first associated to each vertex in a graph. The WDK is then computed as the product of an exact matching kernel over the vertex label with a kernel over the neighborhood edge multiset. The edge label is augmented with the labels of the endpoints. Among the differences between WDK and NSPDK there are: the single vs. pairwise subgraph approach, and the "soft" similarity match vs. the "hard" isomorphism match of neighbourhood subgraphs.



PAIRWISE MAXIMUM COMMON SUBGRAPHS KERNEL (PMCSK)

Schietgat L., Costa F., Ramon J., and De Raedt L.
Maximum common subgraph mining: A fast and effective approach towards feature generation.
In Proc. MLG, pp. 1-3, July 2009.

The PMCSK feature space is obtained considering the maximum common subgraph (MCS) between all pairs of instances in the training set. The authors show that, although the computation of the maximum common subgraph in the general case is an NP-hard problem, one can employ a polynomial-time algorithm if only outerplanar graphs are considered in combination with a special case of subgraph isomorphism called block-and-bridge-preserving (BBP) subgraph isomorphism. In addition to the pairwise vs. single subgraph approach, PMCSK differs from NSPDK in the specific type of subgraphs considered (MCSs vs. neighbourhood graphs).



Example of:
a) non outerplanar graph
b) standard subgraph isomorphism
c) block-and-bridge preserving subgraph isomorphism

Neighbourhood Subgraph Pairwise Distance Kernel

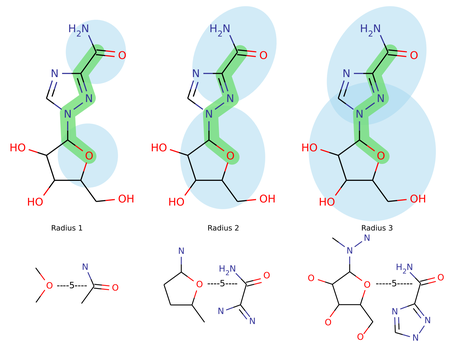
Since the introduction of convolution kernels in (Haussler, 1999), the decomposition approach has been the the guiding principle in kernel design for discrete structured objects. The similarity function is obtained by decomposing each object into (possibly overlapping) parts and by devising a local kernel between the subparts.

For over ten years, machine learning researchers have exploited the remarkable property that it is sometimes possible to efficiently compute this type of kernel, even when objects admit an exponential number of decompositions (e.g. through dynamic programming).

However, as the dimension of the feature space associated with the kernel becomes larger, there is an increasing probability that a significant fraction of the feature space dimensions will be poorly correlated with the target function. As a consequence, even when using large margin classifiers, one can fail to obtain models with good generalization performance (Ben-David et al., 2002).

Possible remedies include down-weighting the contribution of larger fragments, or bounding a priori their size. Alternatively, one can try to identify a strong bias relevant to the task at hand, and consider only a selected subset of structures.

We limit the extracted substructures to pairs of neighbourhood subgraphs, each rooted in one vertex of the given graph and hence efficiently enumerable.



$$\kappa_{r,d}(G,G') = \sum_{\substack{A_v, B_u \in R^{-1}_{r,d}(G) \\ A'_{v'}, B'_{u'} \in R^{-1}_{r,d}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'})$$

$$K(G,G') = \sum_r \sum_d \kappa_{r,d}(G,G').$$

Costa F., De Grave K. Fast neighborhood subgraph pairwise distance kernel.
In Proc. International Conference on Machine Learning (ICML) 2010

How fast?

There is no known algorithm of polynomial complexity for exactly matching graphs (isomorphism). We design a fast approximation with very few collisions.

Encoding neighbourhood balls as graph invariant strings allows hashing

ROOTED GRAPH INVARIANT $\mathcal{L}^g(S)$

1. Compute all pairwise distances $d(u,v)$
2. $\forall u \in V_S$:
   1. $\mathcal{L}^n(u) \leftarrow concat(sort\{(d(u,v), \mathcal{L}(v))\}) \cdot d(u, root)$
3. $\forall (u,v) \in E_S$:
   1. $\mathcal{L}^e(u,v) \leftarrow \mathcal{L}^n(u) \cdot \mathcal{L}^n(v) \cdot \mathcal{L}(u,v)$
4. $\mathcal{L}^g(S) \leftarrow concat(sort\{\mathcal{L}^e\})$

After cryptographic hashing of the features, the kernel can be computed as a sparse vector product.

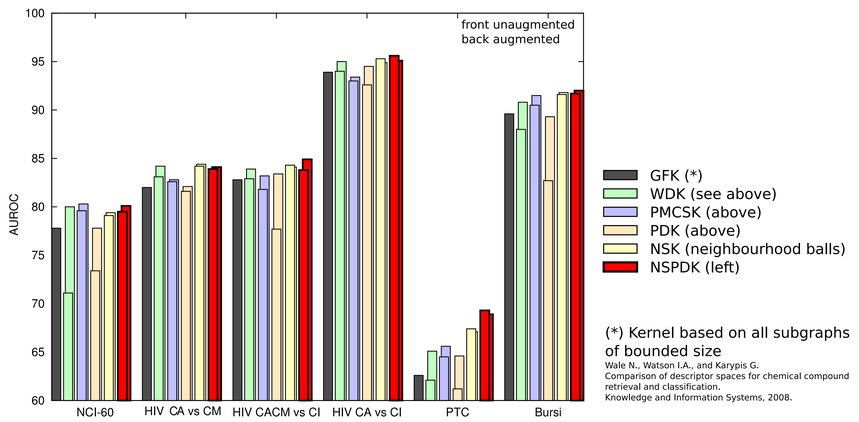
$$\sim 2 \cdot 10^5 \frac{comparisons}{core \cdot second}$$

Table 2. Nett CPU time of graph kernels in seconds

	NCI-60	HIV	PTC	Bursi
# of mol.	3910	42687	417	4337
Aug. time	$3.5 \cdot 10^2$	$3.4 \cdot 10^3$	$3.4 \cdot 10^1$	$1.2 \cdot 10^2$
GFK(G)	$3.5 \cdot 10^1$	$1.4 \cdot 10^4$	$3.1 \cdot 10^0$	$7.3 \cdot 10^1$
WDK(G)	$1.8 \cdot 10^3$	$1.6 \cdot 10^5$	$8.0 \cdot 10^0$	$1.1 \cdot 10^3$
WDK(G_a)	$2.3 \cdot 10^3$	$2.3 \cdot 10^5$	$1.4 \cdot 10^1$	$1.5 \cdot 10^3$
PMCSK(G)	$2.8 \cdot 10^5$	$3.3 \cdot 10^{4*}$	$6.2 \cdot 10^2$	$3.5 \cdot 10^5$
PMCSK(G_a)	$2.8 \cdot 10^5$	$3.3 \cdot 10^{4*}$	$6.3 \cdot 10^2$	$3.5 \cdot 10^5$
PDK(G)	$4.2 \cdot 10^1$	$3.9 \cdot 10^3$	$1.0 \cdot 10^0$	$3.6 \cdot 10^1$
PDK(G_a)	$7.7 \cdot 10^1$	$4.2 \cdot 10^3$	$2.0 \cdot 10^0$	$5.7 \cdot 10^1$
NSK(G)	$6.2 \cdot 10^1$	$3.1 \cdot 10^3$	$2.8 \cdot 10^0$	$5.1 \cdot 10^1$
NSK(G_a)	$3.5 \cdot 10^2$	$6.0 \cdot 10^3$	$1.4 \cdot 10^1$	$2.0 \cdot 10^2$
NSPDK(G)	$1.2 \cdot 10^2$	$1.0 \cdot 10^4$	$3.4 \cdot 10^0$	$1.1 \cdot 10^2$
NSPDK(G_a)	$4.6 \cdot 10^2$	$1.9 \cdot 10^4$	$1.6 \cdot 10^1$	$2.9 \cdot 10^2$

* MCSs derived only from the 1503 CA-CM molecules.

How accurate?



(*) Kernel based on all subgraphs of bounded size
Wale N., Watson I.A., and Karypis G.
Comparison of descriptor spaces for chemical compound retrieval and classification.
Knowledge and Information Systems, 2008.

Conclusions

1. Graph augmentation by and large increases classification performance
2. Increasing the expressiveness of the base kernel, we observe a diminishing return
3. We introduce a fast molecule kernel with excellent generalisation performance

under review by the Journal of Chemical Information and Modeling

Availability

Graph augmentation (free limited license):
http://dtai.cs.kuleuven.be/ml/systems/dmax

NSPDK (GPL source code):
http://dtai.cs.kuleuven.be/ml/systems/nspdk

Acknowledgement

This research was funded by GOA 08/008 "Probabilistic logic learning"